

Attorney Docket No. 826.1767

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Kwame Osei BOATENG

Application No.:

Group Art Unit:

Filed: November 5, 2001

Examiner:

For: APPARATUS AND METHOD FOR TEST-STIMULI COMPACTION

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN  
APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2001-168951

Filed: June 5, 2001

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: November 5, 2001

By: \_\_\_\_\_

James D. Halsey, Jr.  
Registration No. 22,729

700 11th Street, N.W., Ste. 500  
Washington, D.C. 20001  
(202) 434-1500

JC542 U.S. PRO  
09/985768  
11/06/01



PATENT OFFICE  
JAPANESE GOVERNMENT

JC542 U.S. PTO  
09/985768  
11/06/01

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: June 5, 2001

Application Number: Patent Application  
No. 2001-168951

Applicant(s): FUJITSU LIMITED

October 2, 2001

Commissioner,  
Patent Office Kozo Oikawa

Certificate No. 2001-3090000

日本国特許庁  
JAPAN PATENT OFFICE

11/06/01  
1009/985768  
JCG42 U.S. PTO

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2001年 6月 5日

CERTIFIED COPY OF  
PRIORITY DOCUMENT

出願番号

Application Number:

特願2001-168951

出願人

Applicant(s):

富士通株式会社

2001年10月 2日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造

日本特許  
公  
司

【書類名】 特許願  
【整理番号】 0150726  
【提出日】 平成13年 6月 5日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 17/50  
【発明の名称】 テスト刺激コンパクション装置および方法  
【請求項の数】 9  
【発明者】  
【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通  
株式会社内  
【氏名】 クアメ オセイ ボアテン  
【特許出願人】  
【識別番号】 000005223  
【氏名又は名称】 富士通株式会社  
【代理人】  
【識別番号】 100074099  
【住所又は居所】 東京都千代田区二番町8番地20 二番町ビル3F  
【弁理士】  
【氏名又は名称】 大菅 義之  
【電話番号】 03-3238-0031  
【選任した代理人】  
【識別番号】 100067987  
【住所又は居所】 神奈川県横浜市鶴見区北寺尾7-25-28-503  
【弁理士】  
【氏名又は名称】 久木元 彰  
【電話番号】 045-573-3683  
【手数料の表示】  
【予納台帳番号】 012542  
【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 テスト刺激コンパクション装置および方法

【特許請求の範囲】

【請求項1】 デジタル回路のテスト刺激の集合のコンパクションを行うコンパクション装置であって、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択する選択手段と、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除する削除手段と、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力する出力手段と

を備えることを特徴とするコンパクション装置。

【請求項2】 前記出力手段は、前記テスト刺激の集合により検出可能な故障を、テスト刺激を変更することなくカバーするような、該テスト刺激の集合の最小の部分集合を、前記簡単化された集合として出力することを特徴とする請求項1記載のコンパクション装置。

【請求項3】 前記選択手段は、前記部分集合から冗長なテスト刺激を削除した後のテスト刺激の部分集合からエッセンシャルテスト刺激を選択することを階層的に繰り返し、前記出力手段は、選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力することを特徴とする請求項1記載のコンパクション装置。

【請求項4】 前記削除手段は、与えられた故障の集合を最適にカバーするテスト刺激の部分集合を求め、得られたテスト刺激以外の1つ以上のテスト刺激を、前記冗長なテスト刺激として削除することを特徴とする請求項1記載のコンパクション装置。

【請求項5】 前記テスト刺激の集合の情報と、該テスト刺激の集合がカバーする故障の情報と、各テスト刺激を該テスト刺激により検出可能な故障に関係付けるポインティング情報とを格納する格納手段をさらに備え、前記選択手段お

および削除手段は、それぞれ該格納手段に格納された情報を参照して、選択および削除を行うことを特徴とする請求項1記載のコンパクション装置。

【請求項6】 前記テスト刺激の集合を用いて前記デジタル回路のシミュレーションを行い、前記テスト刺激の集合がカバーする故障をトレースするシミュレーション手段をさらに備え、前記格納手段は、各故障に付随して、対応する故障を検出可能なテスト刺激の数を表すカウンタ情報を格納し、前記シミュレーション手段は、前記シミュレーション中に該対応する故障がトレースされたときに該カウンタ情報をインクリメントし、前記選択手段は、該カウンタ情報が1であるような故障をカバーするテスト刺激を、前記エッセンシャルテスト刺激として選択することを特徴とする請求項5記載のコンパクション装置。

【請求項7】 デジタル回路のテスト刺激の集合のコンパクションを行うコンピュータのためのプログラムを記録した記録媒体であって、該プログラムは、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択し、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除し、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力する処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項8】 デジタル回路のテスト刺激の集合のコンパクションを行うコンピュータのためのプログラムであって、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択し、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除し、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力する処理を前記コンピュータに実行させるためのプログラム。

【請求項9】 デジタル回路のテスト刺激の集合のコンパクションを行うコンパクション方法であって、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択し、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除し、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を生成することを特徴とするコンパクション方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、デジタル回路のテストに係り、さらに詳しくはテスト刺激のコンパクションを行う装置および方法に関する。

【0002】

【従来の技術および発明が解決しようとする課題】

信頼できるデジタル／コンピュータシステムの製造を保証するために、システムをテストするテスト刺激 (test stimuli) が生成される。デジタル回路の場合、1つのテスト刺激が单一の入力パターンのときもあり、初期化サブシーケンス (initializing subsequence) 、鋭敏化サブシーケンス (sensitizing subsequence) 、および伝播サブシーケンス (propagation subsequence) のような複数の入力パターンの列のときもある。また、鋭敏化サブシーケンスが伝播サブシーケンスと同じ場合もある。

【0003】

以下では、デジタル回路のテスト刺激を表す用語として、「テストベクトル」が用いられることがある。また、テストベクトルの集合を表す用語としては、「テスト集合」が用いられる。

【0004】

テスト生成の重要な目的は、被テスト回路 (circuit under test) のために生

成されたテストベクトルによる高い故障カバー率（故障検出率）を達成することである。このため、生成アルゴリズムは、必然的に故障カバー率を重視し、冗長性のあるテストベクトルの集合を生成することになる。この冗長性は簡単には検出することができず、結果として大規模なテストベクトルの集合が得られる。

## 【0005】

テスト機器のコストはそのメモリ容量に関係しているため、このような大規模なテストベクトルの集合を用いると、その印加コストが増大する。したがって、テストベクトルの数がテストのコストに強く影響し、信頼できるデジタル／コンピュータシステムの製造と保守のためにそのコストは不可欠となる。メモリ容量は、テスト機器のコストを決定する重要な要素の1つであり、大きなメモリ容量は高いコストにつながる。

## 【0006】

利用可能なテスト機器がすべてのテストベクトルをロードするのに必要なメモリ容量を持っていなければ、被テスト回路へのテスト印加中にテスト機器へのロードを2回以上行う必要がある。つまり、テスト機器のメモリに収容可能な数を超えるテストベクトルの場合、テスト印加中にメモリの更新（テストベクトルの部分集合の再ローディング）が発生する。

## 【0007】

テスト中のテスト機器への再ローディングは、テスト印加時間（*test application time*， TAT）を著しく増大させ、1回のメモリ更新でもテスト印加時間の大幅な増大を招くことになる。長いテスト印加時間は、製造された I C (integrated circuit) 製品の出荷所要時間（time-to-market）に強い影響を与える。

## 【0008】

以上の説明から明らかなように、テストコストを削減し、信頼できるシステムの出荷所要時間を短縮するためには、小規模なテスト集合を用いることが望ましい。したがって、テストベクトルの数を少なくする必要がある。故障カバー率を損なうことなくテストベクトルの数を削減する技術は、テストコンパクション（テスト簡単化）と呼ばれる。テストコンパクションのアルゴリズムは、故障カバー率を低下させることなく小さなテスト集合を得る方法を見つけなければなら

ない。言い換えれば、なるべく高い故障カバー率を保ちながら、テストベクトルの数を少なくしなければならない。

#### 【0009】

コンパクション技術としては、動的コンパクションと静的コンパクションの2種類がある。動的コンパクションは、テストベクトルの生成中にそれらの数を削減する技術であり、しばしば、テスト生成器の変更を必要とする。一方、静的コンパクションは、既に生成されたテストベクトルの数を削減する技術であり、テスト生成の後処理ステップとなる。したがって、静的コンパクションはテスト生成に依存せず、生成アルゴリズムのいかなる変更も必要としない。

#### 【0010】

また、テスト生成で動的コンパクションが用いられたとしても、生成されたテスト集合の大きさを静的コンパクションによりさらに削減することが可能である。このことは、テスト集合の大きさおよび後続テストのコストの削減において、静的コンパクションがより効果的であることを示唆している。

#### 【0011】

Hsiao らは、以下の文献において、順序回路に対するテストシーケンスコンパクションの方法に関する成果について報告している。

- (1) M. S. Hsiao and S. T. Chakradhar, "Partitioning and reordering techniques for static test sequence compaction of sequential circuits," Proc. of the 7th IEEE Asian Test Symposium, pp.452-457, 1998.
- (2) M. S. Hsiao and S. T. Chakradhar, "State relaxation based subsequence removal for fast static compaction in sequential circuits," Proc. of Design, Automation, and Test in Europe Conf., pp.577-582, 1998.
- (3) M. S. Hsiao, E. M. Rudnick and J. H. Patel, "Fast algorithms for static compaction of sequential circuit test vectors," Proc. of IEEE VLSI Test Symposium, pp.188-195, 1997.

彼らの方法は、シーケンスを短くする方法である。一方、Pomeranzらは、以下の文献において、テスト応答の効率的な格納方法を提案している。

- (4) I. Pomeranz and S. M. Reddy, "On test compaction objectives for co

mbinational and sequential circuits," Proc. of IEEE International Conference on VLSI Design, pp.279-284, 1998.

また、KajiharaらとHamzaoglu らは、以下の文献において、組み合わせ回路に対するテストパターンコンパクションに関する成果について報告している。

(5) S. Kajihara and K. Saluja, "On test pattern compaction using random pattern fault simulation," Proc. Of IEEE International Conference on VLSI Design, pp.464-469, 1998.

(6) I. Hamzaoglu and J. H. Patel, "Test set compaction algorithms for combinational circuits," Proc. of ACM International Conference on CAD, pp.283-289, 1997.

彼らは、以前に提案されたコンパクトテスト生成方法に適合する静的コンパクションの方法を開発した。

#### 【0012】

しかし、デジタルシステムには順序回路と組み合わせ回路の両方が含まれ、異なるテストパッケージには異なるテスト生成アルゴリズムが含まれる。したがって、一般的なデジタル回路に対しては、いかなるテスト生成アルゴリズムにもまったく依存しない、一般的な静的テストコンパクションの方法が必要となる。

#### 【0013】

本発明の課題は、一般的なデジタル回路に対して適用可能なテスト刺激コンパクションのための装置および方法を提供することである。

#### 【0014】

##### 【課題を解決するための手段】

図1は、本発明のテスト刺激コンパクション装置の原理図である。図1のコンパクション装置は、選択手段D1、削除手段D2、および出力手段D3を備え、デジタル回路のテスト刺激の集合のコンパクションを行う。

#### 【0015】

選択手段D1は、上記テスト刺激の集合の部分集合から、その部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択する。削除手段D2は、部分集合からエッセンシャルテス

ト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除する。出力手段D3は、選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力する。

## 【0016】

テスト刺激の集合（または、削除手段D2からの部分集合）が与えられると、選択手段D1は、そのテスト刺激の集合から現在特定可能なエッセンシャルテスト刺激の集合を選択して、出力手段D3に渡し、残りのテスト刺激を削除手段D2に渡す。削除手段D2は、選択手段D1から受け取ったテスト刺激の部分集合に含まれる、現在特定可能な冗長なテスト刺激の集合を削除して、残ったテスト刺激を選択手段D1に返す。出力手段D3は、選択手段D1から受け取ったエッセンシャルテスト刺激を、簡単化された集合として出力する。

## 【0017】

このようなコンパクション装置によれば、元のテスト集合により検出可能なすべての故障の検出に不可欠となる、様々な階層のエッセンシャルテスト刺激が、与えられたテスト集合から抽出される。このコンパクション方法は、テスト生成アルゴリズムにまったく依存しないため、いかなる生成アルゴリズムにより生成されたテスト刺激の静的コンパクションに対しても有効である。

## 【0018】

図1の選択手段D1および削除手段D2は、例えば、後述する図16のCPU（中央処理装置）51とメモリ52の組み合わせに対応し、図1の出力手段D3は、例えば、図16のCPU51とメモリ52の組み合わせ、出力装置54、またはネットワーク接続装置57に対応する。また、エッセンシャルテスト刺激は、後述するエッセンシャルテストベクトルに対応する。

## 【0019】

## 【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

本実施形態においては、テスト刺激（テストベクトル）の静的コンパクションのタスクを、最小カバー問題（最小被覆問題）として定式化する。実際には、コンパクション装置は、元のテスト集合により検出可能な対象回路中のすべての故

障を、どのテストベクトルも変更することなく検出（カバー）できるような、元のテスト集合の最小部分集合を求める。コンパクション装置が用いる方法は、情報処理の2つのメインプロセッサの組み合わせで表される。1つはエッセンシャルテストベクトルの選択であり、もう1つは冗長なテストベクトルの削除である。

## 【0020】

また、コンパクション装置は、効率的な実行時間を保証するために、最小カバーの概念に伴うメモリ消費の多い行列の使用を回避できるようなデータ構造を用いる。さらに、最小化問題の解が最小テスト集合であることを保証するために、冗長なテストベクトルを削除する際、ソートアルゴリズムが用いられる。この静的コンパクションの基本概念は、以下の通りである。

定義1：回路Cにおいて、テストベクトルの集合Tsetにより検出可能なすべての故障の集合Fsetは、Tsetのカバーを形成する。TsetによるFsetのカバー状況を表すテーブルは、カバーテーブルと呼ばれる。

定義2：回路Cの故障fがテスト集合Tsetのテストベクトルtepにより検出可能であり、Tsetの他のテストベクトルがfを検出できないとき、tepはTsetのエッセンシャルベクトルであり、故障fはTsetに関する回路Cのエッセンシャル故障である。

## 【0021】

最小カバーの概念は、論理最適化においてしばしば用いられるよく知られたアルゴリズムである。このアルゴリズムには、大きなメモリ容量を必要とするという欠点がある。

## 【0022】

ここで、テストベクトルの集合T = {t<sub>1</sub>, t<sub>2</sub>, t<sub>3</sub>, t<sub>4</sub>, t<sub>5</sub>, t<sub>6</sub>, t<sub>7</sub>} が、回路Cの一部を形成する信号線L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>, L<sub>4</sub>, L<sub>5</sub>, L<sub>6</sub>上のすべてのスタック故障(stuck-at fault)をカバーする場合を考える。対応する故障集合をF = {f<sub>1</sub>, f<sub>2</sub>, f<sub>3</sub>, f<sub>4</sub>, ..., f<sub>11</sub>, f<sub>12</sub>} とし、図2に示すようなカバーテーブルをTのカバーテーブルとする。

## 【0023】

このカバーテーブルにおいて、 $t_i$  の行と  $f_j$  の列の交点の“×”印は、单一故障仮定 (single fault assumption) の下で、テストベクトル  $t_i$  が故障 (fault)  $f_j$  を検出することを表している。この单一故障仮定は、 $f_j$  が被テスト回路内の唯一の故障であるという仮定に相当する。このテーブルからは、 $t_3$ 、 $t_5$ 、 $t_6$ 、 $t_7$  が、エッセンシャル故障  $f_9$ 、 $f_{10}$ 、 $f_{11}$ 、 $f_{12}$  に対応する  $T$  のエッセンシャルベクトルであることが分かる。

## 【0024】

$t_3$ 、 $t_5$ 、 $t_6$ 、 $t_7$  を  $T$  から取り除き、それらがカバーするすべての故障を元のカバー  $F$  から取り除くと、図3に示すような部分カバーテーブルが得られる。このテーブルでは、故障  $f_1$  および  $f_3$  のいずれも、更新されたテスト集合  $T = \{t_1, t_2, t_4\}$  に関してエッセンシャルではない。

## 【0025】

$T$  に残されたテストベクトル  $t_1$ 、 $t_2$ 、 $t_4$  を、残りの故障のカバー状況についてこの順にチェックすると、 $t_4$  が冗長であることが判明する。この冗長なテストベクトル  $t_4$  を廃棄すると、 $f_3$  が  $T = \{t_1, t_2\}$  に関するエッセンシャル故障となり、 $t_2$  が対応するエッセンシャルベクトルとなる。

## 【0026】

そこで、 $T$  からエッセンシャルベクトル  $t_2$  を選択し、それにより検出可能な故障を  $F = \{f_1, f_3\}$  から取り除くと、 $F$  は空になる。したがって、 $T$  に残されたテストベクトル  $t_1$  も廃棄される。こうして、 $t_1$  および  $t_4$  を含まない、簡略化されたテスト集合  $C T = \{t_2, t_3, t_5, t_6, t_7\}$  が得られる。

## 【0027】

一般に、エッセンシャルベクトルの選択と冗長なベクトルの廃棄を数回繰り返すことで、静的コンパクションが効率よく行われる。このコンパクション方法の利点は以下の通りである。

レンマ1：回路  $C$  の与えられた故障集合に対して、エッセンシャルベクトルを持たないいかなるテスト集合  $T$  も、集合  $T_a = T \setminus \{t\}$  に関して冗長な少なくとも1つのテストベクトル  $t$  を有する。

証明：繰り返しのあるサイクルで、テストベクトル  $\tau_1, \tau_2, \dots, \tau_i, \dots, \tau_N$  が残されたとする。これらのテストベクトルの中にエッセンシャルベクトルがなければ、これらのテストベクトルの集合のカバーに含まれ、ベクトル  $\tau_j$  により検出可能いかなる故障も、同じ集合内の他のベクトル  $\tau_k$  ( $k \neq j$ ) により検出可能である。したがって、少なくとも  $\tau_j$  は冗長である。

## 【0028】

Q. E. D.

レンマ1は、エッセンシャルベクトルの選択と冗長なベクトルの廃棄を繰り返し適用する処理において、エッセンシャルベクトルと冗長なベクトルのいずれも存在しない行き詰まりの状況は生じないことを示している。

## 【0029】

ところで、図3に示した残りのデータを処理すると、非最小解に到達する場合も想定される。例えば、 $t_2$  が  $f_3$  のみを検出し、 $t_4$  が  $f_3$  に加えてさらに  $f_1$  を検出するような場合、 $t_1$  と  $t_2$  がエッセンシャルベクトルとして選択され、 $t_4$  が冗長なベクトルとなってしまう。この場合、 $t_4$  だけしか削除されない。このような観点から静的コンパクション問題の最小解を保証するために、次のような処理が用いられる。

## 【0030】

まず、与えられた故障の集合を最適にカバーするテスト刺激の部分集合を求める。そして、得られたテスト刺激以外の1つ以上のテスト刺激を、冗長なテスト刺激として削除する。

## 【0031】

具体的には、エッセンシャルベクトルの集合を選択する度に、残りの  $n$  個のテストベクトルのうち、集合  $F$  内の残りの故障を最も多く検出するものを最初に取り置きする。次に、取り置きされたベクトルにより検出可能な  $F$  内の故障の集合を  $C_1$  として、集合  $S = F \setminus C_1$  内の故障を最も多く検出するテストベクトルを取り置きする。一般には、 $j$  番目に取り置きされたベクトルにより検出可能な  $F$  内の故障の集合を  $C_j$  とし、外1とすると、 $i$  番目に取り置きされるテストベクトルは、

【0032】

【外1】

$$\Gamma = \bigcup_{j=1}^{i-1} C_j$$

【0033】

集合  $S = F \setminus \Gamma$  内の故障を最も多く検出するものである。

このような処理は、 $k$  番目のテストベクトルが取り置きされて  $\Gamma = F$  となるまで続けられる。そして、この時点で残りの  $(n - k)$  個のベクトルは廃棄され、エッセンシャルベクトルの次の階層を選択する新たなサイクルが開始される。

【0034】

また、最小カバーアルゴリズムを直接適用すると、テストベクトルのカバーテーブルを表す行列が必要となる。このため、実用上の回路では、現存しない大容量のメモリが要求されることになり、コンパクション方法の実用上の回路への適用が制限される。この問題を回避するため、行列の2次元データ空間を、以下の方法で、より扱いやすい1次元データ空間に変換する。

【0035】

まず、各々の故障に付随して、その故障を検出可能なテストベクトルの数を表すカウンタを設け、元のテスト集合を用いて対象回路のシミュレーションを行う。このとき、各テストベクトルについて、検出可能な故障をプライマリ出力から逆方向にトレースし、故障がトレースされる度に、その故障に付随するカウンタをインクリメントする。また、テストベクトルを故障にマップ（写像）する。

【0036】

このようなシミュレーションとしては、以下の文献に示された方法を適用することができる。この方法によれば、故障シミュレーションを何度も繰り返すことなく、元のテスト集合によりカバーされる故障の情報を、容易に抽出することができる。

(7) K. O. Boateng, H. Takahashi and Y. Takamatsu, "Diagnosing delay faults in combinational circuits under the ambiguous delay model," IEICE

Transaction on Information and Systems, Vol.E82-D, No.12, pp.1563-1571, Dec. 1999.

(8) K. O. Boateng, H. Takahashi and Y. Takamatsu, "Multiple gate delay fault diagnosis using test-pairs for marginal delays," IEICE Transaction on Information and Systems, Vol.E81-D, No.7, pp.706-715, July 1998.

(9) N. Yanagida, H. Takahashi and Y. Takamatsu, "Multiple fault diagnosis by sensitizing input pairs," IEEE Design & Test of Computers, Vol.12, No.3, pp.44-52, Sept. 1995.

ここで、カウンタの値が1であるようなすべての故障はエッセンシャル故障となり、それらにマップされたそれぞれのテストベクトルはエッセンシャルベクトルとなる。そして、エッセンシャルベクトルが選択される度に、そのベクトルが検出するすべての故障が故障集合から取り除かれる。また、冗長なテストベクトルが特定される度に、更新された故障集合内でそのベクトルがマップされた各故障のカウンタがインクリメントされる。

#### 【0037】

図4は、このようなコンパクション装置により行われる静的コンパクション処理のフローチャートである。この処理は2つのフェーズに分けられる。第1フェーズにおいて、コンパクション装置は、シミュレーションを行って故障をトレースし、ベクトル-故障マッピングを確立する（ステップS1）。故障トレース中には、故障カウンタがインクリメントされる。また、第2フェーズでは、エッセンシャルベクトルの選択処理と冗長なベクトルの廃棄処理を行う（ステップS2）。

#### 【0038】

図5は、第1フェーズの処理のフローチャートである。コンパクション装置は、まず、ベクトル-故障マッピングを確立するためのシミュレーションを行い、故障がトレースされると、それに付随するカウンタをインクリメントする（ステップS11）。次に、すべてのテストベクトルを集合T\_setに入れ、トレースされたすべての故障を集合F\_setに入れる（ステップS12）。そして、集合CT\_set、T1\_set、およびF1\_setを空集合（）に設定する（ステップS13）。

PS13)。ステップS11の処理結果と同様のものがテスト生成により得られている場合、このステップの処理は不要である。

#### 【0039】

図6および図7は、第2フェーズの処理のフローチャートである。コンパクション装置は、まず、現在のすべてのエッセンシャル故障を求める(図6のステップS21)、それらに対応するエッセンシャルベクトルを特定する(ステップS22)。次に、それらのエッセンシャルベクトルにより検出可能なすべての故障をF\_setから削除し(ステップS23)、それらのエッセンシャルベクトルをT\_setからCT\_setに移す(ステップS24)。そして、F\_setが空か否かをチェックする(ステップS25)。

#### 【0040】

F\_setが空であれば、T\_set内のテストベクトルをすべて冗長なテストベクトルとして廃棄し、処理を終了する。F\_setが空でなければ、次に、F\_set内の故障を最も多くカバーするT\_set内のテストベクトルrtを求め(ステップS26)、rtをCT\_setに入れる(ステップS27)。そして、rtにより検出可能な(カバーされる)F\_set内のすべての故障をF1\_setに移し、再びF\_setが空か否かをチェックする(ステップS29)。

#### 【0041】

F\_setが空でなければ、ステップS26以降の処理を繰り返す。F\_setが空であれば、次に、T\_set内にまだ残っている各テストベクトル $\tau$ について、 $\tau$ により検出可能なF1\_set内のすべての故障のカウンタをデクリメントし(図7のステップS30)、T\_set内に残っているすべてのテストベクトルを冗長なテストベクトルとして廃棄する(ステップS31)。そして、F1\_setをF\_setに移して空にし(ステップS32)、T1\_setをT\_setに移して空にして(ステップS33)、図6のステップS21以降の処理を繰り返す。

#### 【0042】

このようなコンパクション処理によれば、コンパクションの終了時には、CT\_set内に簡単化されたテスト集合が格納される。

第1フェーズの処理の後、図2のカバーテーブルのデータは、図8に示すよう

な構成となる。このデータ構造において、 $T\_set = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ 、 $F\_set = \{f_1, f_2, f_3, f_4, \dots, f_{11}, f_{12}\}$  であり、 $T\_set$  内の各テストベクトルから  $F\_set$  内の故障へと向かう矢印は、そのベクトルがそれらの故障をカバーすることを表している。Count = 1 は、エッセンシャル故障およびエッセンシャルベクトルを特定する基準となる。エッセンシャルベクトルからの矢印は太線で記されている。

#### 【0043】

例えば、 $T\_set$ 、 $F\_set$ 、およびカウンタはそれぞれ 1 次元配列に格納され、テストベクトルから故障へと向かう矢印は、 $F\_set$  の配列の対応するインデックスのようなポインティング情報により表される。これにより、テストベクトルが故障に関係付けられる。

#### 【0044】

現在のエッセンシャルベクトルを選択し、それらがカバーするすべての故障を削除して、 $T\_set$  と  $F\_set$  を更新すると、図 9 に示すような新たなマッピングが得られる。残りのテストベクトルは、各ベクトルが残りの故障をいくつ検出可能であるかを調査するために、 $t_1 \rightarrow t_2 \rightarrow t_4$  の順にチェックされる。その結果、 $t_2$  が最大カバーのテストベクトルとして特定される。

#### 【0045】

そこで、 $t_2$  を  $T\_set$  に移し、 $f_1$  と  $f_3$  を  $F\_set$  に移すと、 $F\_set$  が空となる。したがって、 $t_1$  と  $t_4$  が廃棄され、 $t_1$  と  $t_4$  により検出可能な故障である  $f_1$  と  $f_3$  のカウンタがデクリメントされる。更新された  $T\_set$  と  $F\_set$  は、図 10 に示すような構成となり、 $t_2$  は、次に選択されるべきエッセンシャルベクトルとして残される。そして、 $t_2$  を選択し、 $t_2$  により検出可能な  $F\_set$  内の故障を削除すると、 $F\_set$  は空となる。こうして、簡略化されたテスト集合は、 $CT\_set = \{t_2, t_3, t_5, t_6, t_7\}$  となる。

#### 【0046】

図 11 は、被テスト回路の具体例を示しており、図 12 は、この回路のテスト集合を示している。図 12 において、16 個のテストベクトルのパターンは 3 値論理で表されており、“2”は未知論理を表す。テストベクトル  $t_i$  ( $i = 1, 2, \dots, 16$ )

2, . . . , 16) により検出される故障の集合を  $F_{t_i}$  と記し、信号線 L (= 1, 2, 3, . . . , 36, 37, 38) が論理  $\alpha$  (= 0, 1) にスタックする故障 (stuck-at- $\alpha$  fault) を  $L/\alpha$  と記すことになると、各  $t_i$  に対する  $F_{t_i}$  は以下のようになる。

$$F_{t1} = \{1/1 ; 2/0 ; 5/0 ; 6/1 ; 9/0 ; 13/1 ; 15/0 ; 16/0 ; 17/0 ; 18/1 ; 19/0 ; 24/1 ; 25/0 ; 26/0 ; 27/0 ; 29/1 ; 30/1 ; 32/1 ; 33/1 ; 34/1 ; 36/1 ; 38/0\}$$

$$F_{t2} = \{1/0 ; 5/0 ; 6/0 ; 13/0 ; 14/0 ; 15/0 ; 16/0 ; 17/1 ; 18/1 ; 19/1 ; 21/1 ; 24/1 ; 25/1 ; 29/0 ; 32/0\}$$

$$F_{t3} = \{2/1 ; 4/1 ; 10/1 ; 15/1 ; 18/0 ; 24/0 ; 25/0 ; 29/1 ; 32/1\}$$

$$F_{t4} = \{2/0 ; 5/0 ; 10/0 ; 15/0 ; 16/0 ; 18/1 ; 24/1\}$$

$$F_{t5} = \{2/1 ; 3/1 ; 4/0 ; 5/0 ; 8/1 ; 9/1 ; 11/1 ; 14/1 ; 15/0 ; 16/0 ; 17/0 ; 18/1 ; 19/0 ; 22/1 ; 24/1 ; 25/0 ; 27/1 ; 29/1 ; 30/0 ; 32/1 ; 34/0 ; 35/0 ; 36/0 ; 37/0 ; 38/1\}$$

$$F_{t6} = \{1/0 ; 2/1 ; 3/0 ; 4/0 ; 5/0 ; 6/0 ; 9/1 ; 11/0 ; 13/0 ; 14/0 ; 15/0 ; 16/0 ; 17/1 ; 18/1 ; 19/1 ; 21/1 ; 22/1 ; 24/1 ; 25/1 ; 27/1 ; 29/0 ; 30/0 ; 32/0 ; 35/0 ; 36/0 ; 37/0 ; 38/1\}$$

$$F_{t7} = \{4/0 ; 5/0 ; 15/0 ; 16/0 ; 18/1 ; 24/1 ; 36/0 ; 37/0 ; 38/1\}$$

$$F_{t8} = \{5/1 ; 16/1 ; 18/0 ; 24/0 ; 25/0 ; 29/1 ; 32/1\}$$

$$F_{t9} = \{5/0 ; 15/0 ; 16/0 ; 18/1 ; 24/1\}$$

$F_{t10} = \{5/1; 16/1; 18/0; 24/0; 25/0; 29/1; 32/1; 38/0\}$

$F_{t11} = \{1/0; 2/0; 3/1; 5/0; 6/0; 8/0; 12/1; 13/0; 14/0; 15/0; 16/0; 17/1; 18/1; 19/1; 20/1; 21/1; 24/1; 25/1; 26/1; 29/0; 30/0; 32/0; 35/0; 36/0; 37/0; 38/1\}$

$F_{t12} = \{1/1; 2/1; 4/0; 5/0; 6/1; 9/1; 13/1; 15/0; 16/0; 17/0; 18/1; 19/0; 22/1; 24/1; 25/0; 27/1; 29/1; 30/0; 32/1; 34/0; 36/0; 37/0; 38/1\}$

$F_{t13} = \{1/1; 2/0; 5/0; 6/1; 9/0; 13/1; 15/0; 16/0; 17/0; 18/1; 19/0; 24/1; 25/0; 26/0; 27/0; 29/1; 30/1; 32/1; 33/1; 34/1; 36/1; 38/0\}$

$F_{t14} = \{1/0; 2/0; 3/1; 5/0; 6/0; 8/0; 12/1; 13/0; 14/0; 15/0; 16/0; 17/1; 18/1; 19/1; 20/1; 21/1; 24/1; 25/1; 26/1; 29/0; 30/0; 32/0; 35/0; 36/0; 37/0; 38/1\}$

$F_{t15} = \{1/0; 2/0; 3/0; 5/0; 6/0; 7/0; 9/0; 12/0; 13/0; 14/0; 15/0; 16/0; 17/1; 18/1; 19/1; 21/1; 24/1; 25/1; 26/0; 27/0; 28/0; 29/0; 30/1; 31/1; 32/0; 35/1; 37/1; 38/0\}$

$F_{t16} = \{5/1; 16/1; 18/0; 24/0; 25/0; 29/1; 32/1; 38/0\}$

図6に示した第2フェーズの初期段階では、 $t_3, t_4, t_5, t_6, t_{15}$ がエッセンシャルベクトルとなる。これらのエッセンシャルベクトルを選択し、それらが検出する故障を削除すると、残りのデータは図13のようになる。冗長なベクトルの削除処理の初めには、 $F\_set = \{1/1; 5/1; 6/1; 8/0$

; 12/1; 13/1; 16/1; 20/1; 26/1; 33/1; 34/1;  
 36/1} となり、ベクトル  $t_1, t_2, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{16}$  は、それぞれ、残りの故障のうち 6, 0, 0, 2, 0, 2, 4, 3, 6, 4, 2 個を検出する。

## 【0047】

まず、 $t_1$  が  $r t_1$  として選択され、それにより検出される故障の集合は、 $C_1 = \{1/1; 6/1; 13/1; 33/1; 34/1; 36/1\}$  となる。このとき、 $F\_set \setminus C_1 = \{5/1; 8/0; 12/1; 16/1; 20/1; 26/1\}$  となり、ベクトル  $t_2, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{16}$  は、それぞれ、残りの故障のうち 0, 0, 2, 0, 2, 4, 0, 0, 4, 2 個を検出する。

## 【0048】

次に、 $t_{11}$  が  $r t_2$  として選択され、 $C_2 = \{8/0; 12/1; 20/1; 26/1\}$  となる。このとき、 $F\_set \setminus (C_1 \cup C_2) = \{5/1; 16/1\}$  となり、ベクトル  $t_2, t_7, t_8, t_9, t_{10}, t_{12}, t_{13}, t_{14}, t_{16}$  は、それぞれ、残りの故障のうち 0, 0, 2, 0, 2, 0, 0, 0, 2 個を検出する。

## 【0049】

さらに、 $t_8$  が  $r t_3$  として選択され、 $C_3 = \{5/1; 16/1\}$  となる。この段階で、 $F\_set \setminus (C_1 \cup C_2 \cup C_3) = \{\}$  となるので、 $C_1 \cup C_2 \cup C_3 = F\_set$  となる。残りのベクトル  $t_2, t_7, t_9, t_{10}, t_{12}, t_{13}, t_{14}, t_{16}$  の各々について、それがカバーするすべての故障のカウンタをデクリメントした後、これらのベクトルは廃棄される。そして、現在の冗長なベクトルの削除処理が終了する。

## 【0050】

こうして、 $F\_set = \{1/1; 5/1; 6/1; 8/0; 12/1; 13/1; 16/1; 20/1; 26/1; 33/1; 34/1; 36/1\}$  のすべての故障はエッセンシャルとなり、それゆえ  $r t_1 = t_1, r t_2 = t_{11}$ 、および  $r t_3 = t_8$  はエッセンシャルベクトルの 2 番目の階層を形成する。

## 【0051】

これらの3つのベクトルを最初のエッセンシャルベクトルの集合に加えると、元の16個のベクトルのテスト集合により検出可能なすべての故障をカバーする8個のベクトルの集合  $\{t_1, t_3, t_4, t_5, t_6, t_8, t_{11}, t_{15}\}$  が生成される。こうして、コンパクション処理により、故障カバー率を変更することなく、テスト集合は半分の大きさになった。

## 【0052】

以上説明した例では、スタック故障モデルが用いられているが、本発明のコンパクション処理は他の故障モデルにも適用できる。また、組み合わせ回路のテスト刺激の大きさはテストモデルに依存する。そこで、次に、スタック故障モデルと遅延故障 (delay fault) モデルの違いについて説明する。

## 【0053】

図14は、スタック故障モデルの例を示している。ANDゲートG1の出力が0にスタックする故障は、入力パターンにかかわらず出力値を0に設定してしまう。この故障を検出するには、出力を論理1に設定する単一の入力パターン“1, 1”が印加される。もし予想される論理1が出力として観測されれば、故障は存在せず、逆に出力値が0になれば、故障の存在が確認される。このように、スタック故障をテストするには、1つの入力パターンのテスト刺激が必要である。

## 【0054】

図15は、遅延故障モデルの例を示している。遅延故障とは、ゲートの入力に発生した変化が出力に伝わるのに、予想より長い時間を要するような状況に相当する。遷移遅延故障 (transition delay fault) モデルでは、遅延故障は遷移と時間の組み合わせで表される。この場合、ゲート出力には、立ち下がり遅延 (slow-to-fall) と立ち上がり遅延 (slow-to-rise) の2つの故障が起こり得る。

## 【0055】

ANDゲートG2の出力の立ち上がり遅延故障を検出するには、まず、“0, 1”のような第1のパターンにより出力を論理“LOW”（論理0）に設定する。次に、第2のパターン“1, 1”を印加した後に所定時間（δユニット）経過したところで、出力論理を“HIGH”に変更する（0から1に立ち上がらせる

) ことを試みる。

【0056】

もし第2のパターンの印加からδユニット経過した後に出力論理が1に変われば、故障は存在せず、そうならなければ、遅延故障が検出されることになる。このように、単一の遅延故障をテストするには、2つの入力パターンからなるテスト刺激が必要となる。

【0057】

また、コンパクション方法を順序回路に適用するには、初期化サブシーケンス、銳敏化サブシーケンス、および伝播サブシーケンスの列を単一のテスト刺激として扱えばよい。

【0058】

ところで、本実施形態のコンパクション装置は、例えば、図16に示すような情報処理装置（コンピュータ）を用いて構成することができる。図16の情報処理装置は、CPU（中央処理装置）51、メモリ52、入力装置53、出力装置54、外部記憶装置55、媒体駆動装置56、およびネットワーク接続装置57を備え、それらはバス58により互いに接続されている。

【0059】

メモリ52は、例えば、ROM (read only memory)、RAM (random access memory) 等を含み、コンパクション処理に用いられるプログラムとデータを格納する。CPU51は、メモリ52を利用してプログラムを実行することにより、必要な処理を行う。

【0060】

入力装置53は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、ユーザからの指示や情報の入力に用いられる。出力装置54は、例えば、ディスプレイ、プリンタ、スピーカ等であり、ユーザへの問い合わせや処理結果の出力に用いられる。

【0061】

外部記憶装置55は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク装置、テープ装置等である。情報処理装置は、この外部記憶装置55に、

上述のプログラムとデータを保存しておき、必要に応じて、それらをメモリ52にロードして使用する。

【0062】

媒体駆動装置56は、可搬記録媒体59を駆動し、その記録内容にアクセスする。可搬記録媒体59としては、メモリカード、フロッピーディスク、CD-R OM (compact disk read only memory)、光ディスク、光磁気ディスク等、任意のコンピュータ読み取り可能な記録媒体が用いられる。ユーザは、この可搬記録媒体59に上述のプログラムとデータを格納しておき、必要に応じて、それらをメモリ52にロードして使用する。

【0063】

ネットワーク接続装置57は、LAN (local area network) 等の任意の通信ネットワークに接続され、通信に伴うデータ変換を行う。また、情報処理装置は、上述のプログラムとデータをネットワーク接続装置57を介して他の装置から受け取り、必要に応じて、それらをメモリ52にロードして使用する。

【0064】

図17は、図16の情報処理装置にプログラムとデータを供給することができるとコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体59やサーバ60のデータベース61に保存されたプログラムとデータは、メモリ52にロードされる。このとき、サーバ60は、プログラムとデータを搬送する搬送信号を生成し、ネットワーク上の任意の伝送媒体を介して情報処理装置に送信する。そして、CPU51は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【0065】

(付記1) デジタル回路のテスト刺激の集合のコンパクションを行うコンパクション装置であって、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択する選択手段と、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集

合から、冗長なテスト刺激を削除する削除手段と、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力する出力手段と

を備えることを特徴とするコンパクション装置。

(付記2) 前記出力手段は、前記テスト刺激の集合により検出可能な故障を、テスト刺激を変更することなくカバーするような、該テスト刺激の集合の最小の部分集合を、前記簡単化された集合として出力することを特徴とする付記1記載のコンパクション装置。

(付記3) 前記選択手段は、前記部分集合から冗長なテスト刺激を削除した後のテスト刺激の部分集合からエッセンシャルテスト刺激を選択することを階層的に繰り返し、前記出力手段は、選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力することを特徴とする付記1記載のコンパクション装置。

(付記4) 前記削除手段は、与えられた故障の集合を最適にカバーするテスト刺激の部分集合を求め、得られたテスト刺激以外の1つ以上のテスト刺激を、前記冗長なテスト刺激として削除することを特徴とする付記1記載のコンパクション装置。

(付記5) 前記テスト刺激の集合の情報と、該テスト刺激の集合がカバーする故障の情報と、各テスト刺激を該テスト刺激により検出可能な故障に関係付けるポインティング情報とを格納する格納手段をさらに備え、前記選択手段および削除手段は、それぞれ該格納手段に格納された情報を参照して、選択および削除を行うことを特徴とする付記1記載のコンパクション装置。

(付記6) 前記テスト刺激の集合を用いて前記デジタル回路のシミュレーションを行い、前記テスト刺激の集合がカバーする故障をトレースするシミュレーション手段をさらに備え、前記格納手段は、各故障に付随して、対応する故障を検出可能なテスト刺激の数を表すカウンタ情報を格納し、前記シミュレーション手段は、前記シミュレーション中に該対応する故障がトレースされたときに該カウンタ情報をインクリメントし、前記選択手段は、該カウンタ情報が1であるような故障をカバーするテスト刺激を、前記エッセンシャルテスト刺激として選択することを特徴とする付記5記載のコンパクション装置。

(付記7) スタック故障モデルの故障を検出可能なテスト刺激の集合のコンパクションを行うことを特徴とする付記1記載のコンパクション装置。

(付記8) 遅延故障モデルの故障を検出可能なテスト刺激の集合のコンパクションを行うことを特徴とする付記1記載のコンパクション装置。

(付記9) 初期化サブシーケンス、銳敏化サブシーケンス、および伝播サブシーケンスの列を单一のテスト刺激とみなして、前記テスト刺激の集合のコンパクションを行うことを特徴とする付記1記載のコンパクション装置。

(付記10) デジタル回路のテスト刺激の集合のコンパクションを行うコンピュータのためのプログラムを記録した記録媒体であって、該プログラムは、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択し、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除し、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力する処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

(付記11) デジタル回路のテスト刺激の集合のコンパクションを行うコンピュータのためのプログラムであって、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択し、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除し、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力する処理を前記コンピュータに実行させるためのプログラム。

(付記12) デジタル回路のテスト刺激の集合のコンパクションを行うコンピュータにプログラムを搬送する搬送信号であって、該プログラムは、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により

検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択し、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除し、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を出力する処理を前記コンピュータに実行させることを特徴とする搬送信号。

(付記13) デジタル回路のテスト刺激の集合のコンパクションを行うコンパクション方法であって、

前記テスト刺激の集合の部分集合から、該部分集合内の他のテスト刺激により検出不可能な少なくとも1つの故障を検出する、エッセンシャルテスト刺激を選択し、

各部分集合からエッセンシャルテスト刺激を選択した後のテスト刺激の部分集合から、冗長なテスト刺激を削除し、

選択されたエッセンシャルテスト刺激からなる簡単化された集合を生成することを特徴とするコンパクション方法。

#### 【0066】

##### 【発明の効果】

本発明によれば、テスト刺激の静的コンパクションを最小カバー問題として扱うことで、テスト生成アルゴリズムに依存しないテスト刺激コンパクションが実現される。このコンパクションは、一般的なデジタル回路の任意のテスト生成アルゴリズムにより生成されたテスト刺激に対して有効である。

##### 【図面の簡単な説明】

###### 【図1】

本発明のテスト刺激コンパクション装置の原理図である。

###### 【図2】

テスト集合のカバーテーブルを示す図である。

###### 【図3】

テスト集合の部分カバーテーブルを示す図である。

###### 【図4】

コンパクション処理のフローチャートである。

【図5】

第1フェーズの処理のフローチャートである。

【図6】

第2フェーズの処理のフローチャート（その1）である。

【図7】

第2フェーズの処理のフローチャート（その2）である。

【図8】

最初のベクトル故障マッピングを示す図である。

【図9】

更新されたベクトル故障マッピングを示す図である。

【図10】

さらに更新されたベクトル故障マッピングを示す図である。

【図11】

デジタル回路の例を示す図である。

【図12】

デジタル回路のテスト集合を示す図である。

【図13】

部分カバーテーブルの概要を示す図である。

【図14】

スタック故障モデルを示す図である。

【図15】

遅延故障モデルを示す図である。

【図16】

情報処理装置の構成図である。

【図17】

記録媒体を示す図である。

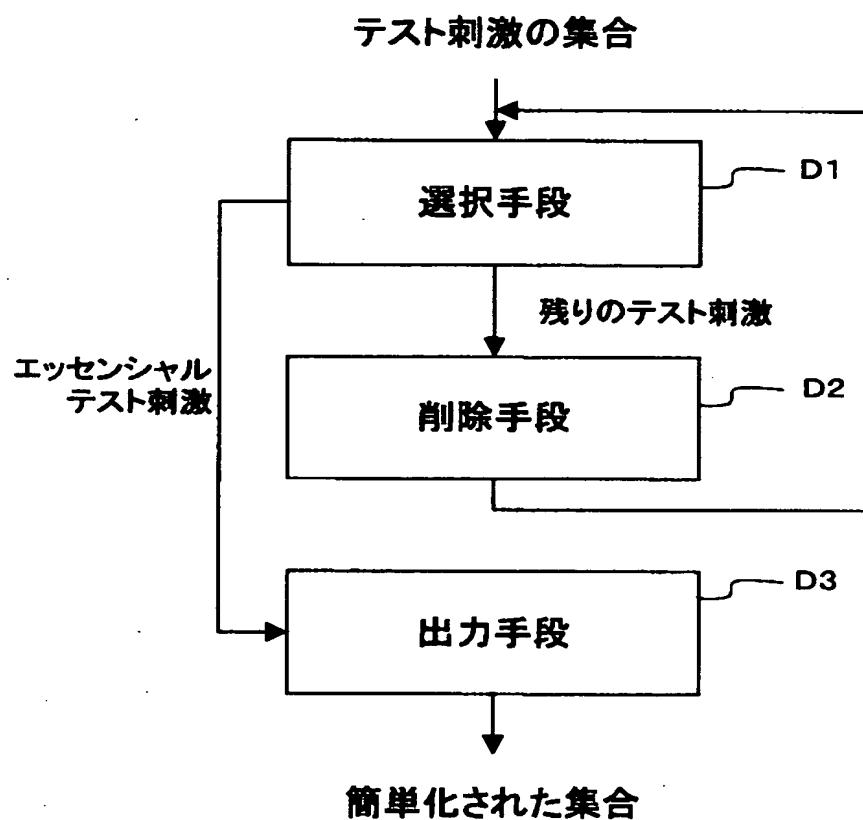
【符号の説明】

1～4 1 信号線

- 5 1 CPU
- 5 2 メモリ
- 5 3 入力装置
- 5 4 出力装置
- 5 5 外部記憶装置
- 5 6 媒体駆動装置
- 5 7 ネットワーク接続装置
- 5 8 バス
- 5 9 可搬記録媒体
- 6 0 サーバ
- 6 1 データベース
- D 1 選択手段
- D 2 削除手段
- D 3 出力手段

【書類名】 図面  
【図1】

## 本発明の原理図



【図2】

## テスト集合のカバーテーブルを示す図

Test Vector \ Fault	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$
Test Vector	$t_1$	×	×	×			×					
	$t_2$	×		×	×					×		
	$t_3$		×			×	×	×			×	
	$t_4$			×			×					
	$t_5$				×				×			
	$t_6$				×			×	×			×
	$t_7$					×	×	×				×

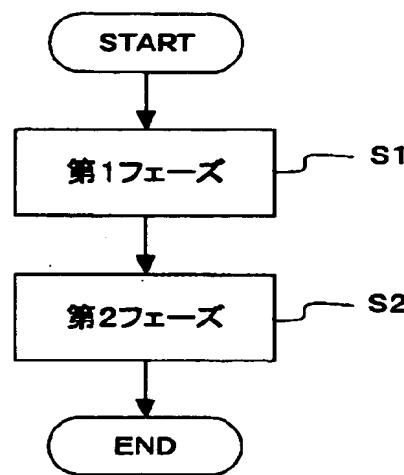
【図3】

## テスト集合の部分カバーテーブルを示す図

Test Vector \ Fault	$f_1$	$f_3$
Test Vector	$t_1$	
	$t_2$	×
	$t_3$	×
	$t_4$	×

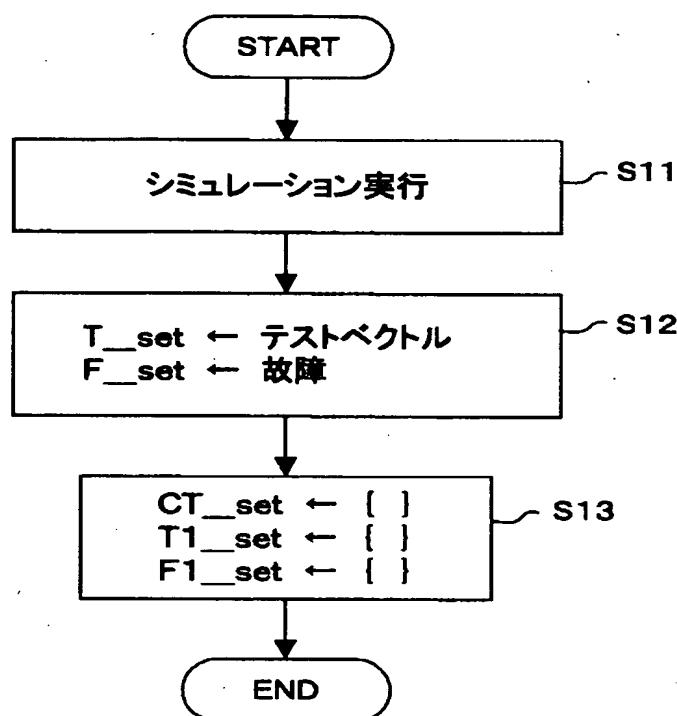
【図4】

コンパクション処理のフローチャート



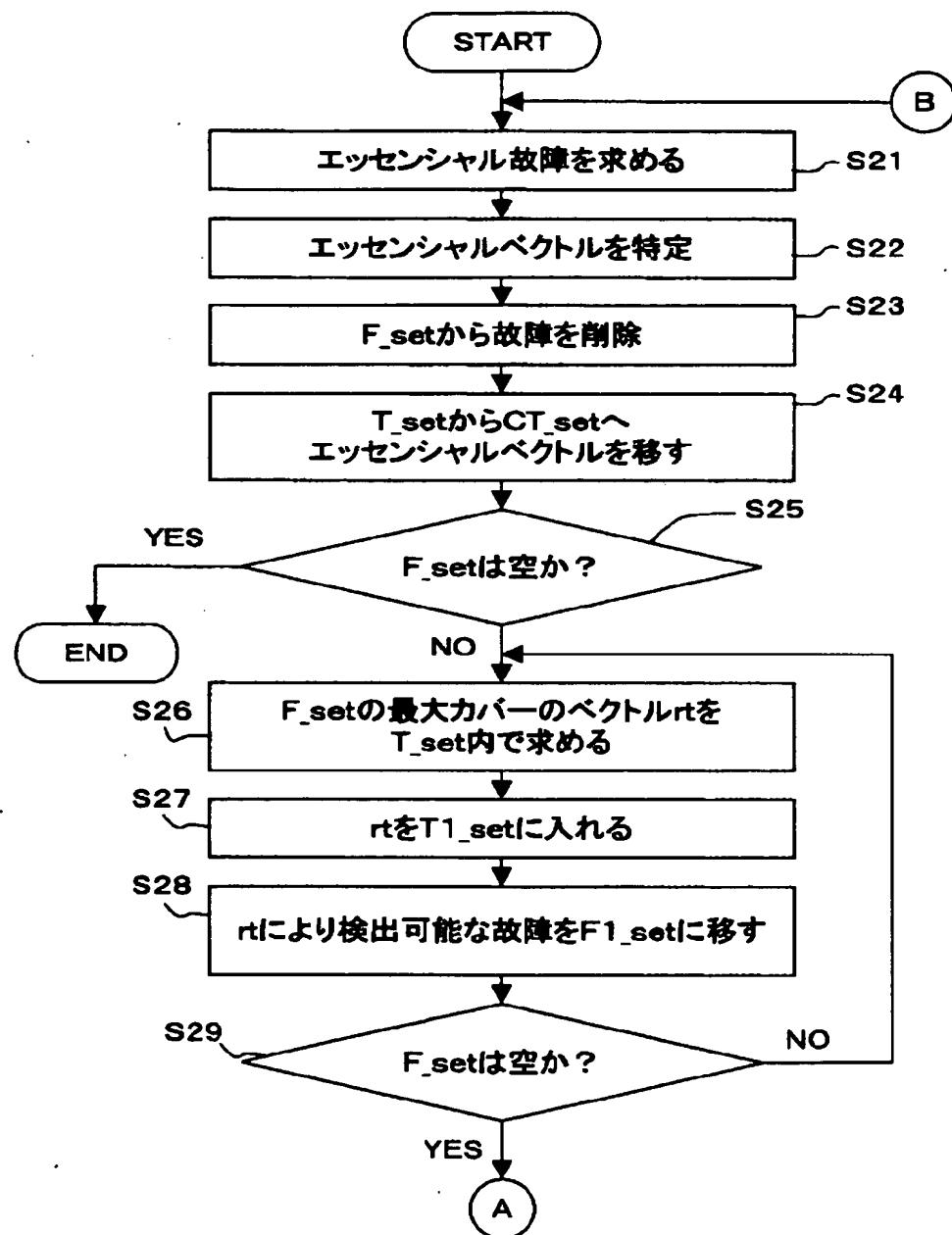
【図5】

## 第1フェーズの処理のフローチャート



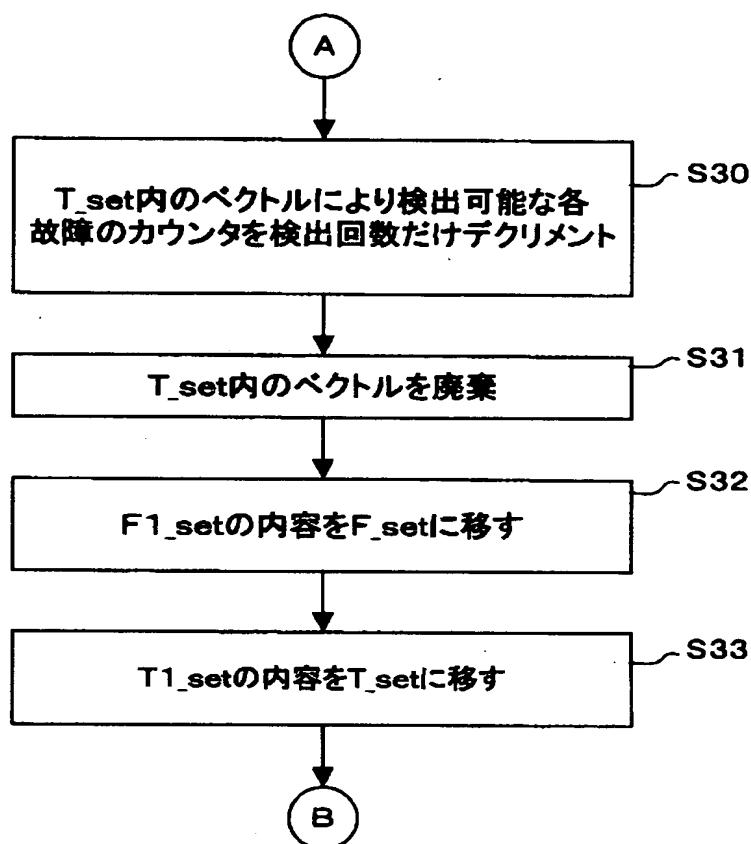
【図6】

## 第2フェーズの処理のフローチャート(その1)



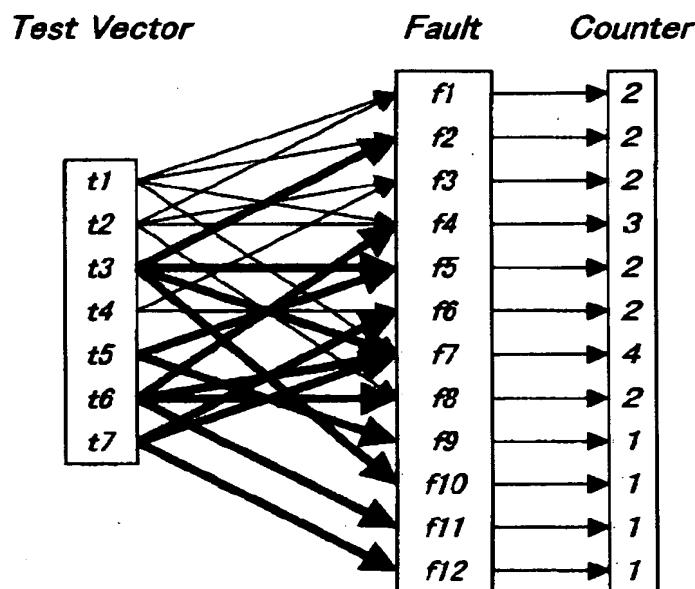
【図7】

## 第2フェーズの処理のフローチャート(その2)



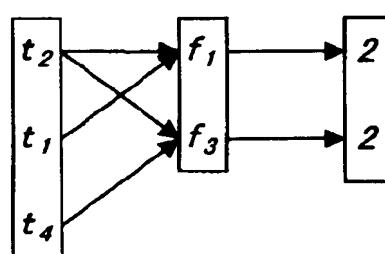
【図8】

## 最初のベクトルー故障マッピングを示す図



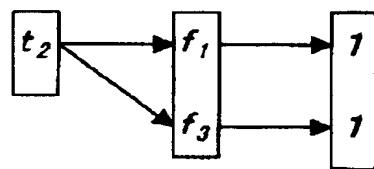
【図9】

## 更新されたベクトルー故障マッピングを示す図



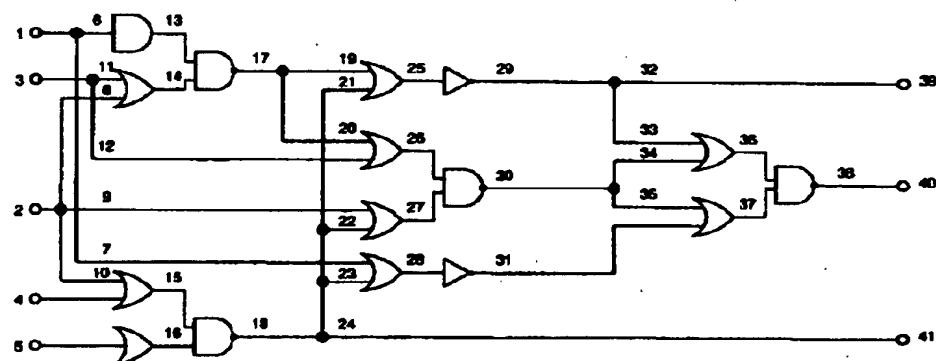
【図10】

さらに更新されたベクトル一故障マッピングを示す図



【図11】

デジタル回路の例を示す図



【図12】

## デジタル回路のテスト集合を示す図

Test Vector	Pattern
$t_1$	01221
$t_2$	11221
$t_3$	20201
$t_4$	21201
$t_5$	10011
$t_6$	10111
$t_7$	20211
$t_8$	21220
$t_9$	21221
$t_{10}$	10010
$t_{11}$	11011
$t_{12}$	00111
$t_{13}$	01111
$t_{14}$	11021
$t_{15}$	11121
$t_{16}$	01020

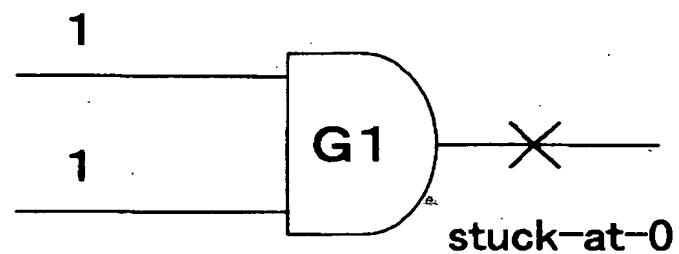
【図13】

## 部分カバーテーブルの概要を示す図

Fault	Counter	Test Vector	#Fault
1/1	3	$t_1$	6
5/1	3	$t_2$	0
6/1	3	$t_7$	0
8/0	2	$t_8$	2
12/1	2	$t_9$	0
13/1	3	$t_{10}$	2
16/1	3	$t_{11}$	4
20/1	2	$t_{12}$	3
26/1	2	$t_{13}$	6
33/1	2	$t_{14}$	4
34/1	2	$t_{16}$	2
36/1	2		

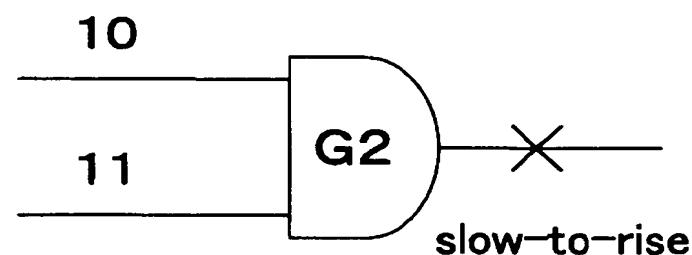
【図14】

### スタック故障モデルを示す図



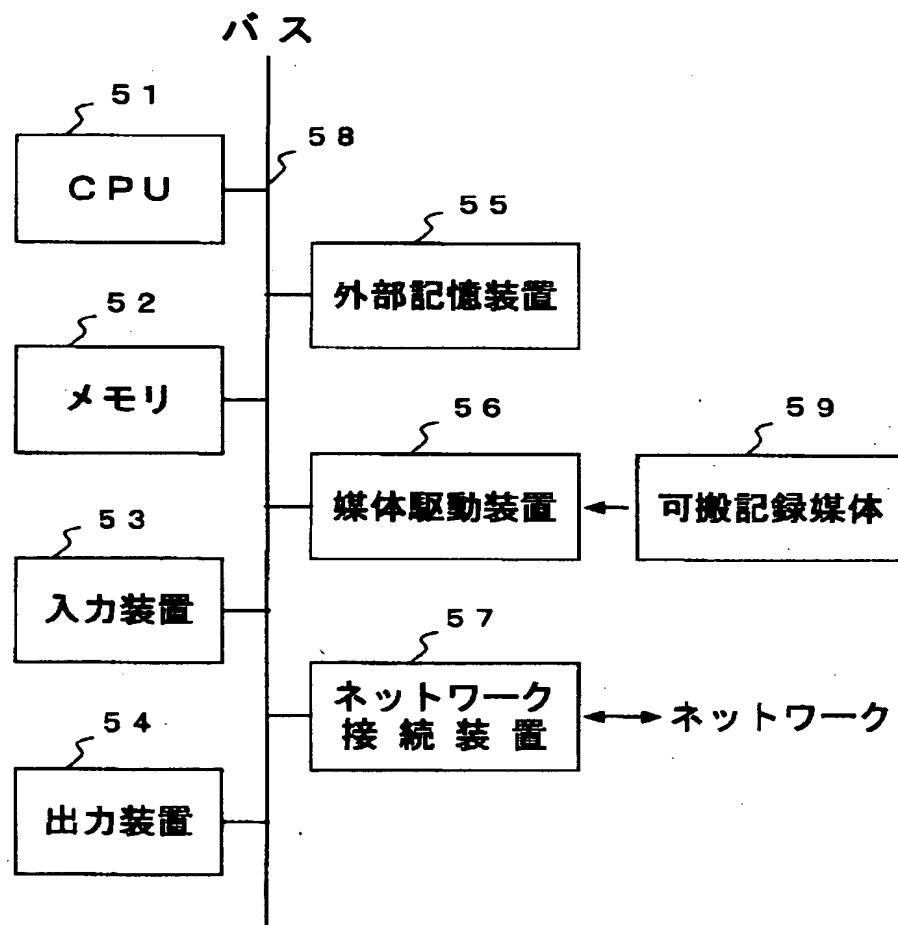
【図15】

### 遅延故障モデルを示す図



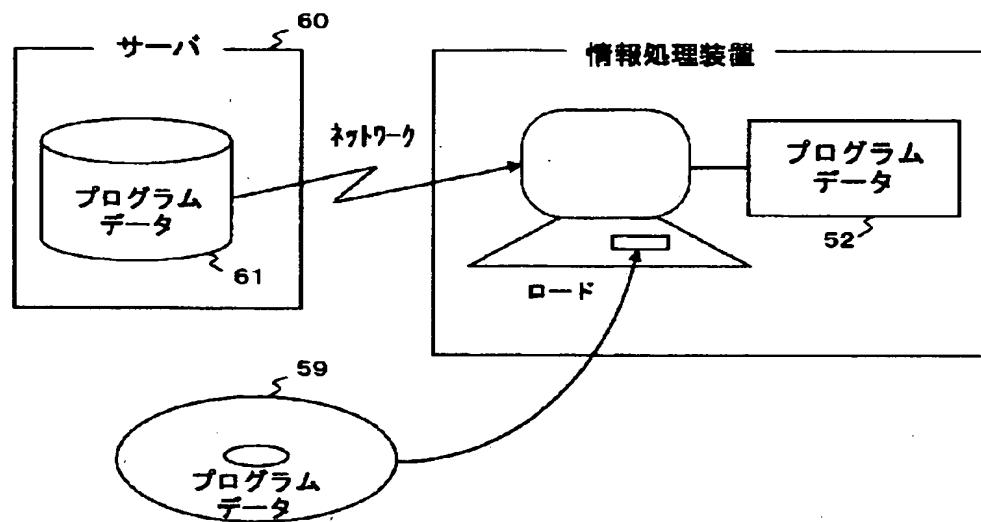
【図16】

## 情報処理装置の構成図



【図17】

## 記録媒体を示す図



【書類名】 要約書

【要約】

【課題】 一般的なデジタル回路に対して適用可能なテスト刺激コンパクションを提供することが課題である。

【解決手段】 テストベクトルの集合は故障の集合にマップされ、各故障には、それを検出可能なテストベクトルの数を表すカウンタが設けられる。カウンタが1である故障にマップされたテストベクトル  $t_3, t_5, t_6, t_7$  は、エッセンシャルベクトルとして選択され、残りのテストベクトルについてマッピングが更新される。そして、次に、 $t_2$  がエッセンシャルベクトルとして選択され、残りの冗長なテストベクトル  $t_1$  および  $t_4$  は廃棄される。

【選択図】 図8

出願人履歴情報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号

氏 名 富士通株式会社